

Basics of C

Storage Classes in C

Riya Jana

April 16, 2024

Introduction of Storage Classes

Definition:

- In C programming language, storage classes specify the lifetime, scope and visibility of variables.

What is lifetime, scope and visibility?

```
#include<stdio.h>
```

```
Int main()
```

```
{  
    int x;  
    x=sum();  
    printf("x: %d", x);  
    return 0;  
}
```

```
int sum()
```

```
{  
    int a=5, b=10, c;  
    c=a+b;  
    return c;  
}
```

Block of code where
lifetime, scope and
visibility came to
know.

Types of Storage Classes

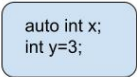
There are four types of storage classes.

- Automatic Storage Class(used as auto or default)
- Register Storage Class(used as register)
- Static Storage Class(used as static)
- External Storage Class(used as extern)

Automatic Storage Class

- These type of storage class variable are initialised with some garbage value until it explicitly initialized.
- It allocates the memory in runtime.

```
#include<stdio.h>
Int main()
{
    auto int x;
    int y=3;
}
```

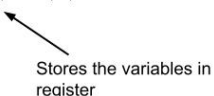


Both are doing same job in memory

Register Storage Class

- The register storage class is similar to the automatic storage class, used to define local variables that should be stored in a CPU register instead of RAM for faster access.
- It is used to optimized the performance.
- Compiler ignores the register keyword because nowadays compiler can take better decisions about register allocations than programmers, knowing everything about hardware.

```
int sum()
{
    register int a=5, b=10, c;
    c=a+b;
    return c;
}
```



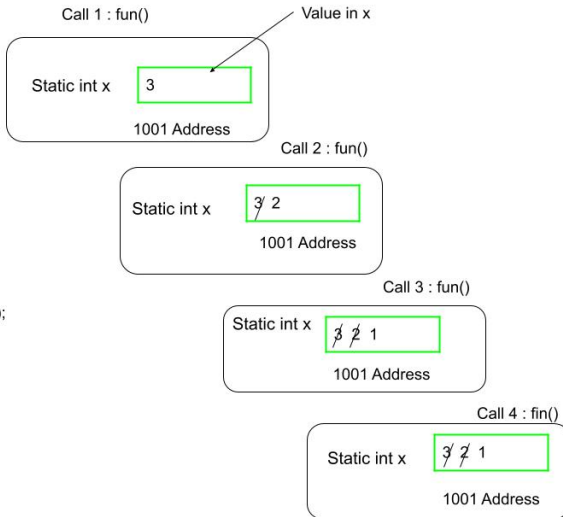
Stores the variables in register

Static Storage Class

- The static storage class is used to declare static variables, which are initialized only once and retain their value between function calls.
- Once it created it stays lifetime until program terminates. It change its value with every function call if needed.
- It allocates the memory at compile time.

Example: I

```
#include<stdio.h>
void fun();
int main()
{
    fun();
}
void fun()
{
    static int x=3;
    x--;
    printf("hellow\n");
    if(x > 0)
    {
        fun();
        printf("bye\n");
    }
}
```



Example:II

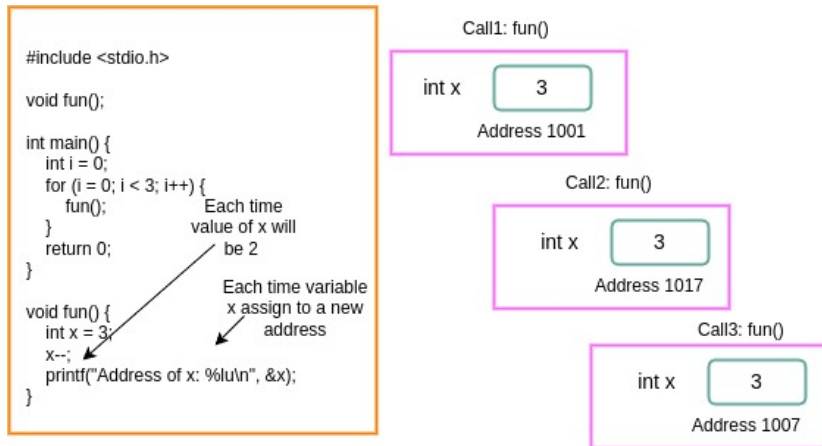


Figure: An example image

External Storage Class

- The external storage class is used to declare variables that are defined in another source file or globally.
- It's useful when you have multiple files sharing variables/functions.
- the extern keyword is used to extend the visibility of variables/functions.
- It initialize the value in runtime.

Example

Tells compiler there is a variable x which need to be access from outside

\\ File1.c

```
#include <stdio.h>
extern int x;
int main()
{
    printf("The value of x is: %d\n", x);
    return 0;
}
```

To fetch value of x need to access file.c

x

10	
----	--

\\File.c

```
int x=10;
```

[illegible]